Supplementary Materials for Bootstrapping Autonomous Driving Radars with Self-Supervised Learning

Yiduo Hao[*,†]
University of Cambridge

Sohrab Madani[†]
UIUC

Junfeng Guan
EPFL

Mohammed Alloulah
RadarEye

Saurabh Gupta
UIUC

Haitham Hassanieh
EPFL

1. Data Compression

Motivation. Section 4.4.2 of the main paper explains how we apply antenna dropouts and random phase noise to an intermediate 3-D complex tensor to for data augmentation during the self-supervised training. However, this requires that we load this data into memory, which takes orders of magnitude more memory, significantly slowing down the data loading phase. To counter this, we compress the radar data as we elaborate in this section.

We first provide a more detailed explanation of the radar processing pipeline. To create radar heat maps, a raw radar heat map is processed through a pipeline to get a $L \times A$ range-azimuth map in the end. Right before the MIMO antennas are combined, an intermediate variable calculated in the process, which we denote by x, is a complex 4-D tensor of shape (M, N, L, A). Here L represents range indices, A the azimuth indices. M and N also represent the number of transmitters and receivers respectively in the MIMO setup. Sometimes x is reshaped into the 3-D shape (MN, L, A). The range azimuth map, which we denote by x_{RA},is achieved as follows:

$$x_{\text{RA}}(\rho, \theta) = \left| \sum_{m,n}^{M,N} x(m, n, \rho, \theta) \right|,$$

where the sums over m and n aggregate all the antenna channel, giving us the (L, A) range-azimuth map. However, as mentioned above, the augmentations in Section 4.4.2 of the main paper require loading x into memory, which is MN times larger than the x_{RX} ins size, on top of being complex-valued, which makes it in take $2MN\times$ space. For example, in the *Radatron* dataset, the range azimuth map generation involves 86 effective antennas, this translates to $172\times$ larger memory requirement. We therefore opt for a data compression method that we explain below.

[*]Work done during internship at EPFL.
[†]denotes co-primary first authors.

Compression Method. The methodology involves decomposing a 3-D complex tensor x of dimensions (MN, L, A) into its magnitude and angle components for separate compression. Representing x as $r \odot \exp(i\theta)$, where r and θ are 3-D tensors of identical dimensions, and $\odot$ signifies element-wise multiplication, we proceed with the compression process. This involves linearly quantizing θ into N_θ levels and logarithmically quantizing r into N_r levels.

The linear quantization of θ can be expressed as:

$$Q_\theta(x) = \left\lfloor \frac{\theta - \theta_{\text{min}}}{\Delta_\theta} \right\rceil \Delta_\theta + \theta_{\text{min}},$$

where $\Delta_\theta = \frac{\theta_{\text{max}} - \theta_{\text{min}}}{N_\theta - 1}$ and $\theta_{\text{min}}, \theta_{\text{max}}$ are the minimum and maximum values of θ, respectively.

The logarithmic quantization of r can be formulated as:

$$Q_r(x) = \exp\left(\left\lfloor \log\left(\frac{r}{r_{\text{min}}}\right) \cdot \frac{N_r}{\log\left(\frac{r_{\text{max}}}{r_{\text{min}}}\right)} \right\rceil \cdot \frac{\log\left(\frac{r_{\text{max}}}{r_{\text{min}}}\right)}{N_r} \right) \cdot r_{\text{min}},$$

where r_{min} and r_{max} are the minimum and maximum values of r.

In *Radical*, we choose $N_\theta = N_r = 256$, which compresses each complex number to 2 bytes, a 16-fold compression compared to `double` and 8-fold compared to `single`. We found little to no difference in training accuracy using the compressed version of the data.

2. Results

Here we present additional results on top of the main results of the paper.

Orientation Split. We show our method's performance against random initialization for different car orientations, following *Radatron* [3]. Table 1 shows the results for *Radical* against *Radatron* (with random initialization) for *straight*, *oriented*, and *incoming cars*. Straight car are those on the same lane as the ego-vehicle, and have roughly

Method	AP	AP_{50}	AP_{75}	AP_{str}	AP_{ori}	AP_{inc}
Radatron [3]	56.5 ± 0.2	88.9 ± 0.4	64.5 ± 1.7	61.9 ± 0.4	32.9 ± 0.7	30.9 ± 0.9
Radical (ours)	**62.3**$\pm$0.6	**89.6**$\pm$0.1	**69.7**$\pm$1.2	**68.7**$\pm$0.5	**33.0**$\pm$1.5	**31.8**$\pm$1.3
Vision Labels + finetune	59.3 ± 0.9	89.0 ± 0.3	67.8 ± 0.7	65.4 ± 1.2	32.8 ± 1.2	30.8 ± 0.6

Table 1. **Comparison of more settings and with vision label pretraining** For row 1, the backbone of the detection model is randomly initialized. For row 2 and 3, we pre-trained the model with 32k Radatron unlabeled frames, and fine-tuned it on 13k Radatron labeled frames. Results are averaged for 6 runs.

Eval Metric		AP 50 (%)				AP 75 (%)				mAP (%)			
Model	Split	str.	ori.	inc.	overall	str.	ori.	inc.	overall	str.	ori.	inc.	overall
Radatron [3]		94.0	59.1	69.5	88.9	72.1	35.2	25.0	64.5	61.9	32.9	30.9	56.5
Radical (intra)		94.2	60.2	70.5	89.0	75.2	34.7	25.3	66.8	65.5	32.9	31.4	59.4
Radical (cross)		94.5	58.7	72.1	89.3	75.8	33.2	25.2	67.1	65.8	32.3	32.2	59.7
Radical (intra+cross)		**94.5**	**58.7**	**73.0**	**89.6**	**78.5**	**31.4**	**27.2**	**69.8**	**68.9**	**31.2**	**32.9**	**62.3**

Table 2. **Extra Granular Results** For row 1, the backbone of the detection model is randomly initialized. For row 2, 3 and 4, we pre-trained the model with 32k Radatron unlabeled frames, and fine-tuned it on 13k Radatron labeled frames. Results are averaged for 6 runs.

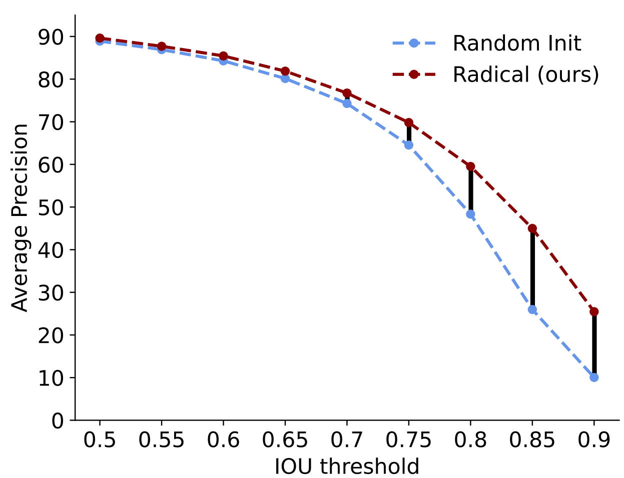

Figure 1. **Average Precision plotted against the IOU threshold for Random Initialization vs. *Radical*.** The plot demonstrates a growing disparity in performance as the IOU threshold increases.

the same direction. Incoming cars are those on the opposite lane, and have roughly the opposite direction. Oriented cars are all those in between, such as cars directed towards the right or left while making a turn. As seen from Table 1, the most significant portion of *Radical*'s gain comes from *straight* cars. This shows two things; first, there is large room for improvement regarding the detection of common scenarios like straight cars on the road using radars. In this paper, this is achieved through self-supervised training. Second, high-resolution radar struggles to accurately detect incoming and oriented cars, even with the help of pre-training. Therefore, future efforts aiming at significantly improving radar performance should specifically tackle multipath and secularity in radar, since these artifacts are the main reasons behind the performance of AP_{ori} and AP_{inc} lagging behind.

Vision Labels Baseline. In order to compare with pseudo-labels that are vision-generated, we create another baseline to compare *Radical* against and show its results in the last line of Table 1. In this baseline, we pre-train the model by training on the whole 63k frames of the dataset with vision-based pseudo-labels. The vision-based pseudo-labels are generated by a Stereo RCNN model. The pre-trained network is then fine-tuned with ground-truth human-generated labels. Although this approach provides some gain (2.8% in mAP), *Radical* outperforms this method by 3% in mAP. In addition to poorer performance, the vision-based labels from Stereo RCNN model require careful calibration and projection from vision to radar, while this is not a requirement by *Radical*'s method. Finally, generating reliable vision labels in the bird's eye view requires multiple (at least two) time-synchronized cameras. This is not required by *Radical*.

Performance boost against IOU thresholds. We mentioned in sec. 6 of the main paper that the gain from *Radical* mostly comes from improving the details, leading to higher gains for AP_{75} compared to AP_{50}. Here we present another result that further confirms our assertion. Specifically, we compare *Radical*'s average precision performance with that of supervised training with random initialization, for different IOU thresholds. Fig. 1 shows the results. As demonstrated, the gap in performance between *Radical* and the baseline increases as the IOU threshold goes up. At 0.5 threshold, the difference is a mere 0.3% improvement for *Radical*. However, this increases to more than 10% for a threshold of 0.8, and more than 15% for thresholds of 0.85 and 0.9. This further demonstrates that *Radical* significantly improves over the baseline in more challenging scenarios where a higher Intersection Over Union (IOU)

Eval Metric		AP 50 (%)				AP 75 (%)				mAP (%)			
Model	Split	str.	ori.	inc.	overall	str.	ori.	inc.	overall	str.	ori.	inc.	overall
Radatron [3]	original split	93.5	84.0	78.2	91.1	50.8	39.3	38.3	47.8	51.8	43.0	40.7	49.4
Radical(ours)	original split	94.3	83.4	77.2	91.5	60.3	32.5	32.3	54.6	55.6	40.0	37.5	52.1

Table 3. **Results in the original Radatron dataset split [3]** For row 1, the backbone of the detection model is randomly initialized. For row 2, we pre-trained the model with 32k Radatron unlabeled frames, and fine-tuned it on 13k Radatron labeled frames. Results are averaged for 6 runs.

λ_{intra}	0.2	1	5	20
mAP	62.1 ± 0.5	62.3 ± 0.6	61.7 ± 1.2	59.9 ± 0.3

Table 4. **Results for the hyper-parameter λ_{intra}** in *Radical* settings.

threshold is required. This improvement is indicative of *Radical*'s ability to refine object detection with greater precision, particularly in scenarios where a more exact overlap between the predicted and ground truth bounding boxes is necessary.

Dataset train/test split. We also present the results for *Radatron* and *Radical* with the original dataset split as in [3] in Table 3. We find significant boosts in performance by *Radical* for straight car conditions, especially in AP 75. However, we observe that the performance for oriented and incoming cars are dropped. This might be caused by the biases in the 32k pre-training dataset, which lacks oriented and incoming cars scenarios. The hyper-parameters used are also tuned for the new dataset train/test split. The lower performance in oriented cars can also be further analyzed by changing the backbone architecture[4]. We would also like to mention that the overall variance for the results in this dataset split is much higher.

Other vision encoders. We also tried a ImageNet[1] pre-trained ViT[2] as the image encoder, yielding 62.1 ± 0.7 mAP, slightly lower than CLIP image encoder. This shows the wide applicability of our method.

Other hyper-parameters. We also ablated the hyper-parameter λ_{intra}. Experiments show that $\lambda_{\text{intra}} = 1$ work the best

3. Implementation Details

Contrastive learning objective choice. All the results presented in the paper are done in SimCLR-style contrastive learning objective. Thus, we did not use a momentum encoder or a large negative sample queue for the results in our paper. While the 64 batch size is not large, it proved to work at the same performance level as a MoCo-like queue-based implementation in our cross-modal setting experiments (64 batch size and 4k negative queue). We believe that this is due to the sparsity of radar heatmaps and the size of the dataset. In fact, changing the batch size from 64 to 8k for a MoCo-like objective gives similar performance results.

4. Additional Qualitative Results

We show additional *randomly sampled* qualitative results samples from our test set in Fig. 2. We also compare *Radical*'s performance against *Radatron*.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[3] Sohrab Madani, Junfeng Guan, Waleed Ahmed, Saurabh Gupta, and Haitham Hassanieh. Radatron: Accurate detection using multi-resolution cascaded mimo radar. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, page 160–178, 2022. 1, 2, 3

[4] Yifan Pu, Yiru Wang, Zhuofan Xia, Yizeng Han, Yulin Wang, Weihao Gan, Zidong Wang, Shiji Song, and Gao Huang. Adaptive rotated convolution for rotated object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6589–6600, 2023. 3

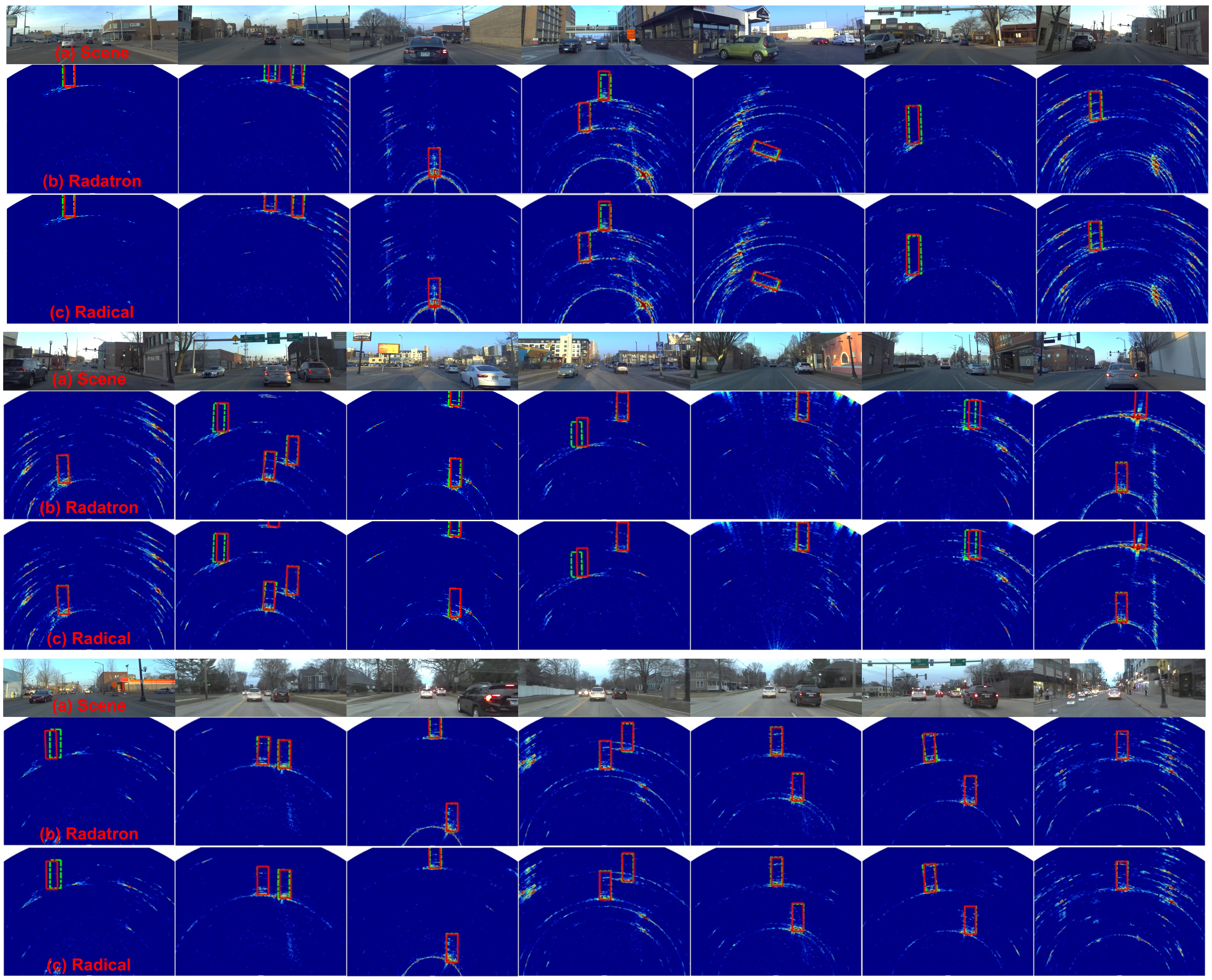

Figure 2. **Randomly sampled examples from our test set.** (a) Original scene. (b) *Radatron* (supervised) baseline. (c) *Radical*. Groundtruth marked in green and predictions in red.